verification data from cookies residing on the end user's system or a combination of requested data and cookie data. The end user continues in step 140 by navigating through Web pages on the information provider's Web site until the desired information is located. During this process, the end user is often required to visit Web pages of little or no use to the end user whose goals is to simply acquire the particular PI residing on the Web site. Ultimately in step 150, the end user is presented with the desired PI. The entire process 100 is repeated for each individual piece of PI desired by the end user. Under this PI access model, the end user must visit each separate information provider, track potentially different identity verification data for each, utilize a different user interface at each site and possibly wade through a significant number of filler Web pages.

Please replace the paragraph on page 10, lines 14 - 23 with the following:

In addition, or as an alternative, the PI associated with each end user 210 may reside on his/her client computer 220 using cookie technology as specified in D. Kristol and L. Montulli, "HTTP State Management Mechanism", Request For Comments (RFC) 2109, February, 1997 (available at http://www.ietf.org/rfc/rfc2109.txt), which is expressly incorporated herein in its entirety. The PI associated with the end user 210 would be stored as PI cookies 375. This implementation mechanism provides inherent support for segregating PI associated with one end user 375 from PI associated with all other end users. Utilizing this method as a substitute for a centralized store provides a layer of security against unauthorized access. As a further measure, PI data stored in cookies could be stored in an encrypted format.

Please replace the paragraphs at page 12, line 11 – page 15, line 16 with the following:

The four primary processing components access and manipulate the data in the three stores. The processing components may execute on a single processor, such as

a file server computer system based on a Pentium class (MMX, PRO, II, III, etc.) central processing unit or an equivalent, or multiple processors. These four processing components are the Baseline configure component 320, the end user configure component 330, the PI access/transact component 340 and the PI delivery component 350 as seen in FIG. 3. The Baseline configure component 320 provides the interface by which new user selectable PI providers are added to the system. This component 320 might be implemented in a variety of ways including trial and error followed by manual entry of configuration information, semi-automated trial and error (automated location of Hypertext Markup Language (HTML) <FORM> elements, Javascript functions and Java applets) followed by manual entry of configuration information or, preferably, configuration by example (executing the protocol in a simulated Web client where the simulated Web client automatically generates a list of required data and a list of steps in the access process). These processes would be utilized at two levels: the first level being the set of data and steps required for general access to the particular PI provider and the second level being the set of additional data and steps required for accessing each particular piece of PI on the PI provider's site. The baseline configuration component 320 may be triggered independently when a new PI provider is added to the system, or it might be triggered as a result of a failure of the PI access/transact component 340 potentially indicating a change in access requirements for the failed access. This latter warning would more likely result where the PI access/transact component 340 has made a comparison between requirements supplied by the Provider store 310, both general to the PI provider and specific to the PI or transaction, and the end user data supplied by the user store 360 after seeking end user verification via a request of the end user to confirm the previously entered required access data via the end user configure component 330 and found an inconsistency. When an inconsistency is determined, updates to the Provider store 310 are made to bring the Provider data into conformance with current access/transaction requirements.

The end user configure component 330 allows an end user to select and configure PI and transactions of interest to the specific user. This configuration information is maintained in the user store 360. When an end user initially subscribes to the system according to the present invention, the system allows the user to select the types and sources of PI and/or transactions desired. First, the system requests permission from the end user to act on his behalf to obtain any selected PI and to execute any authorized transactions. Next, the system provides the user with a list of known information suppliers and the types of PI supplied from and transactions supported by the particular PI provider from the Provider store 310. The system requests the verification data necessary for accessing each selected PI provider and the additional data required by the particular PIs and/or transactions desired from that PI provider. Assuming the end user is already a registered user with the selected PI provider or the particular PI provider does not require prior registration, the data supplied by the end user is placed in the user store 360.

One method of obtaining any cookie data would be for the end user to access each previously accessed PI utilizing the PI engine 240 as a proxy server. The PI engine 240 would pass the cookie data to the PI provider site with the appropriate Web page requests to obtain the PI or execute the transaction and with the end user's permission retain a copy of the cookie data in the end user's record in the user store 360. An alternate means of obtaining the cookie data would be a direct upload of the cookie information from the end user's computer. In a preferred embodiment, no cookie data is necessary where a user is already registered with a provider. All that is necessary is the verification data for login.

If the end user does not have the requisite information because he is not a registered user of a selected PI provider, the user configure component 330 prompts the user for the information necessary to register the end user with the PI provider and performs the registration procedure required by the PI provider. A simulated Web client could perform this process automatically supplying the access data as required and

sending any necessary cookie data. The manner in which such a simulated client registers the end user depends significantly upon the interaction method used on the PI provider Web site. If the Web site uses HTML forms and common gateway interface (CGI) applications, the end user configure component 330 can formulate a uniform resource locator (URL) to replicate the effect of actual form usage and submit this URL to the simulated Web client. The use of a URL to mimic an HTML form is equivalent to manually entering the data into the Web <FORM> element. See Kerven, Foust, Zakour, HTML 3.2 Plus How-To, Waite Group Press, 1997, pp. 559-569. If the Web site uses a mixture of HTML forms and Javascript functions, a simulated Web client with a modified Javascript interpreter could effectively register the user by following the end user registration process for the particular PI provider. The registration process to follow would be obtained from the record of the particular PI provider in the Provider store 310. The Javascript interpreter in the simulated Web client would follow this procedure and supply the data supplied by the end user. A similar process could be used if the registration process on the PI provider Web site utilizes a Java applet. A Web client with a modified Java bytecode interpreter could effectively register the user by following the end user registration process stored for the particular PI provider in the Provider store 310. The bytecode interpreter would supply the data previously entered by the end user rather than requiring interactive input from the end user. If the PI provider Web site utilizes a combination of forms, scripts and applets, the individual procedures above could be used in combination to accomplish the desired registration.

Please replace the paragraph at page 17, lines 1 – 19 with the following:

A failed registration could result from several situations. First, the end user attempting to register with the PI provider does not qualify for registration; for example, an end user attempting to register with a bank with whom the end user does not maintain an account and where the bank only allows access to account holders. Next, the end user may have supplied improper or incorrect information. For example, a bank

registration process might require a social security number, a password, a bank account number and the maiden name of the end user's mother; if the user entered an incorrect social security number, the registration process would fail. Finally, the PI provider may have altered the registration procedure for its Web site. In this situation, following the process supplied from the Provider store 310 would yield a failed registration. In the instance of any registration failure, the end user could be presented with the data initially supplied to the system for registration. The system could then ask the end user to double check the correctness of the information provided and to correct and resubmit the data if an error is found. A second failure resulting from the submission of identical requisite data might generate an error message presented to the end user stating that either the end user is ineligible to access the selected PI from the selected PI provider or that alteration by the PI provider may have caused an error in registration. This second failure could also trigger a warning suggesting the need to potentially reconfigure the record for the PI provider in the Provider store 310.

Please replace the paragraphs on page 18, lines 20 – page 20, line 14 with the following:

With reference to FIG. 3, the PI access/transact component 340 supports the update, acquisition and transaction functionality of the PI engine 240. The PI access/transact component 340 is responsible for accessing and storing user PI and executing transactions authorized by the end user. When access or update is needed for a selected end user, the PI access/transact component 340 combines information from the Provider store 310 and the user store 360 to update end user PI in the PI store 280. For each piece of PI requiring access or update, the PI access/transact component 340 looks up the access procedure and information needed for the particular PI in the Provider store 310. The verification and access data is found in the user store 360. The PI access/transact component 340 utilizes this information to connect to the PI provider's Web site across the Internet and to access the PI. Where

multiple pieces of PI require updating or access, the accesses may occur in series or parallel.

Requested transactions would be similarly supported. For each transaction, the PI access/transact component 340 combines information from the Provider store 310 and the user store 360 to perform the requested transaction. The PI access/transact component 340 looks up the transaction procedure and information needed for the particular transaction in the Provider store 310. The verification and access data is found in the user store 360. The PI access/transact component 340 utilizes this information to perform the transaction across the Internet from the PI provider's Web site.

A simulated Web client could perform access or transaction processes automatically supplying access and verification data as necessary. The manner in which such a simulated client access PI or execute transactions depends significantly upon the interaction method used on the PI provider Web site. If the Web site uses HTML forms and common gateway interface (CGI) applications, the PI access/transact component 340 can formulate a uniform resource locator (URL) to replicate the effect of actual form usage and submit this URL to the simulated Web client. The use of a URL to mimic an HTML form is equivalent to manually entering the data into the Web <FORM> element. See Kerven, Foust, Zakour, HTML 3.2 Plus How-To, Waite Group Press, 1997, pp. 559-569. If the Web site uses a mixture of HTML forms and Javascript functions, a simulated Web client with a modified Javascript interpreter could effectively access the PI or perform the transaction by following the PI access/transact process for the particular PI or transaction respectively. The access or transaction process to follow would be obtained from the record of the particular PI or transaction in the Provider store 310. The Javascript interpreter in the simulated Web client would follow this procedure and supply the data found in the user store 360. A similar process could be used if the PI provider Web site utilizes a Java applet. A Web client with a modified Java bytecode interpreter could effectively access PI or perform transactions by

following process stored for the particular PI or transaction in the Provider store 310. The bytecode interpreter would supply the data from the user store 360 rather than requiring interactive input from the end user. If the PI provider Web site utilizes a combination of forms, scripts and applets, the individual procedures above could be used in combination to accomplish the desired access.

Please replace the paragraph at page 29, lines 7 – 15 with the following:

The end user must first identify the Provider 110. Next, the end user must locate the Provider's Web address 120. Then, the user requests the Provider's login page 130. If the end user does not remember the requisite information, this information must be found, or the desired information will remain inaccessible via the Web. The end user then navigates the Provider's Web site 140. This often entails visiting the Provider's main page 710 followed by viewing a variety of intermediate pages on the Provider's site 720. The end user may have to backtrack several times to the main page 710 or accidentally leave the system entirely forcing a second login 130 before finally locating the desired information 150.

Please replace the paragraph at page 32, lines 12 – 21 with the following:

For instance, with reference to FIG. 2 an end user 210 would be able to maintain his/her accounts online through the PI Engine 240. If an information provider has the capability of receiving payments online, the PI Engine 240 could support complete or partial automation of such transactions. If there is a billing due date for a certain information provider, PI Engine 240 could flag that information and send email to the end user 210 notifying him/her of the bill due. Thus, the user will not have to check each of his/her providers individually for due date information. The PI Engine 240 could also automated payments on a limited range of billing amount for providers who allow payments over their Web servers 250, then send an email to the user with the notification of payment.